

**UTILITY  
PATENT APPLICATION  
TRANSMITTAL**

Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No.

9392-025-999

Total Pages

54

First Named Inventor or Application Identifier

STUART A. KAUFFMAN

Express Mail Label No.

N/A

**APPLICATION ELEMENTS**

See MPEP chapter 600 concerning utility patent application contents.

ADDRESS TO:

Assistant Commissioner for Patents  
Box Patent Application  
Washington, DC 20231

1. ☒ Fee Transmittal Form  
*Submit an original, and a duplicate for fee processing)*
2. ☒ Specification [Total Pages 42]  
*(preferred arrangement set forth below)*
- Descriptive title of the Invention
  - Cross Reference to Related Applications
  - Statement Regarding Fed sponsored R&D
  - Reference to Microfiche Appendix
  - Background of the Invention
  - Brief Summary of the Invention
  - Brief Description of the Drawings *(if filed)*
  - Detailed Description of the Invention *(including drawings, if filed)*
  - Claim(s)
  - Abstract of the Disclosure

3. ☒ Drawing(s) *(35 USC 113)* [Total Sheets 8]
4. ☒ Oath or Declaration [Total Sheets 2]

- a. ☐ Newly executed (original or copy)
- b. ☐ Copy from a prior application (37 CFR 1.63(d))  
*(for continuation/divisional with Box 17 completed)*

**[Note Box 5 below]**

- i. ☐ DELETION OF INVENTORS(S)

Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33 (b).

5. ☐ Incorporation By Reference *(useable if Box 4b is checked)*  
The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Box 4b, is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference therein.

6. ☐ Microfiche Computer Program *(Appendix)*
7. ☐ Nucleotide and/or Amino Acid Sequence Submission  
*(if applicable, all necessary)*
- a. ☐ Computer Readable Copy
  - b. ☐ Paper Copy (identical to computer copy)
  - c. ☐ Statement verifying identity of above copies

**ACCOMPANYING APPLICATION PARTS**

8. ☐ Assignment Papers (cover sheet & document(s))
9. ☐ 37 CFR 3.73(b) Statement ☒ Power of Attorney  
*(when there is an assignee)*
10. ☐ English Translation Document *(if applicable)*
11. ☐ Information Disclosure Statement (IDS)/PTO-1449 ☐ Copies of IDS Citations
12. ☐ Preliminary Amendment
13. ☒ Return Receipt Postcard (MPEP 503)  
*(Should be specifically itemized)*
14. ☐ Small Entity Statement filed in prior application, Statement(s) Status still proper and desired
15. ☐ Certified Copy of Priority Document(s)  
*(if foreign priority is claimed)*
16. ☐ Other:

17. If a **CONTINUING APPLICATION**, check appropriate box and supply the requisite information:  
☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No:

**18. CORRESPONDENCE ADDRESS**☒ Customer Number or Bar Code Label

20582

*(Insert Customer No. or Attach bar code label here)*or ☐ Correspondence address below

NAME

ADDRESS

CITY

STATE

ZIP CODE

COUNTRY

TELEPHONE

FAX

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231.

# A METHOD AND SYSTEM FOR RESOURCE ALLOCATION

## FIELD OF THE INVENTION

5 The present invention relates generally to a method and system for resource allocation. More particularly, the present invention allocates resources using technology graphs, passive and active searching, reinforcement learning, market driven decision making, reinforcement learning as well  
10 as *p*, *tau*, and *patches* techniques.

## Background

Modern computer networks for enterprises are tending towards the distributed computer system model in which various resources such as data, application programs,  
15 and computational power, are available at different locations on a network. Commercial products (e.g., Sun's Enterprise Java Beans, and Microsoft's COM, COM+ and MTS systems) and standards (CORBA) exist that are directed at providing a common interface for diverse applications that will allow  
20 platform-independent operation.

For heterogenous components to work together successfully across an enterprise, the Object Management Group (OMG) adopted guidelines for how component interfaces are defined and components are invoked. Recently, OMG has  
25 been actively soliciting proposals from vendors for the development of an architecture and a series of detailed specifications for CORBA. The CORBA core includes standardized network communication protocols, interface mappings for various languages, interoperability, interface  
30 and implementation repositories, and calling conventions implemented by a CORBA Object Request Broker (ORB).

An ORB is the middleware that established the client-server relationship between objects. Using an ORB, a client can invoke the method of a server object transparently. This server object can be on the same machine  
5 as the client object or across the network. The ORB intercepts the call and is responsible for finding an object that can implement the request, passing it data, invoking the requested operation and returning the results.

Microsoft has been promoting its own alternative to  
10 CORBA for a distributed software component architecture. The Microsoft approach is based on Microsoft's COM. COM principally provides the architectural abstractions for Object Linking and Embedding (OLE) components. COM was based on several fundamental concepts: as application binary  
15 interface (ABI) for calling functions, a provisions for grouping functions into interfaces and a base interface. COM defined a standard for laying out virtual functional tables in memory and a standard for calling functions through pointers.

20 Microsoft has developed the Microsoft Transaction Server (MTS) to extend COM to service other levels of brokering that may be required in an application. MTS is a component-based transaction processing systems for developing and deploying transaction-oriented processing on Windows NT  
25 Server. MTS also defines an application programming model for developing distributed, component-based applications.

But each of these products is limited to providing a common interface for diverse application to allow platform-independent operation. Accordingly, there exists a need for algorithms that adaptively discover resource availability and  
30 balance the load among a set of resource providers.

### Summary of the Invention

The present invention is a method and system to adaptively discover resource availability from resource providers and to adaptively balance the load among the  
5 resource providers.

The present invention includes a method for resource allocation comprising the steps of:

receiving a plurality of resource requests and a plurality of resource offers, said resource requests and said  
10 resource offers having one or more properties; and

determining at least one relation between at least one of said resource requests and at least one of said resource offers to identify at least one matching said resource offer for said resource request; and

15 allocating said at least one resource request with said at least one matching resource offer.

The present invention further includes a method for allocating a plurality of resources comprising the steps of:

receiving information for at least one resource  
20 request;

computing an expected return for processing said resource request from said information; and

processing said at least one resource request to optimize said expected return.

25 The present invention further includes a method for allocating a plurality of resources comprising the steps of:

defining at least one algorithm having one or more parameters for allocating the resources;

defining at least one global performance measure of  
30 said at least one algorithm;

executing said algorithm for a plurality of different values of said one or more parameters to generate a

corresponding plurality of values for said global performance measure;

- constructing a fitness landscape from said values of said parameters and said corresponding values of said
- 5 global performance measure; and
- optimizing over said fitness landscape to generate optimal values for said at least one parameter.

## 10 Brief Description of Drawings

FIG. 1 shows an illustration of the architecture of the system of the present invention.

FIG. 2 provides a flow diagram describing a method executed by the resource providing nodes.

- 15 FIG. 3 displays an exemplary technology graph.

FIG. 4 provides a dataflow diagram 400 representing an overview of a method for synthesizing the technology graph.

- FIG. 5 provides a flow diagram for locating and
- 20 selecting *poly-functional intermediate objects* for a set of terminal resources having a cardinality greater than or equal to two.

- FIG. 6 displays a flow diagram of a method for allocating resources using a market-based scheme which could
- 25 also execute on a resource providing node.

FIG. 7 provides a flow diagram for determining optimal values of parameters of methods performing resource allocation and load balancing.

- FIG. 8 discloses a representative computing system
- 30 in conjunction with which the embodiments of the present invention may be implemented and executed.

### Detailed Description of the Preferred Embodiment

The present invention allocates resources using technologies graphs, passive and active searching,  
5 reinforcement learning, market driven decision making, reinforcement learning as well as *p*, *tau*, and *patches* techniques.

Without limitation, the following embodiments of the present invention are described in the illustrative  
10 context of the allocation of resources in a distributed, computing system. However, it will be apparent to persons of ordinary skill in the art that other contexts can be used to embody the aspects of the present invention. These aspects, which are applicable in a wide range of contexts include  
15 receiving a plurality of resource requests and a plurality of resource offers, determining at least one relation between the resource requests and the resource offers to identify matching resource requests and offers and allocating the resource offers to its matching resource request.

20

#### **System Architecture**

FIG. 1 shows an illustration of the architecture of the system of the present invention. The system includes resource requests 110, 112, 114, 116, which could be  
25 generated by a client application. The resource requests 110, 112, 114, 116 have one or more properties. Exemplary properties for the resource requests 110, 112, 114, 116 include identities and requirements. Exemplary properties for the resource requests 110, 112, 114, 116 further include attributes such as *is-a*, *needs-a*, *has-a*, *uses-a*, etc.  
30 Exemplary attributes for resource requests further include requirements for data, software, computational or

communication resources at a specified level of quality of service.

The system of the present invention further includes resource offers 130, 132, 134, 136, which could be  
5 generated by resource providers. The resource offers 130, 132, 134, 136 have one or more properties. Exemplary properties for the resource offers 130, 132, 134, 136 include identities and abilities. Exemplary properties for the resource offers 130, 132, 134, 136 further include attributes  
10 such as is-a, has-a, needs-a, does-a, etc. Exemplary attributes for resource offers 130, 132, 134 further include data, software, computational or communication resource abilities at a specified level of quality of service.

The system of the present invention further includes resource providing nodes (RPNs) 120, 122, 124, 126,  
15 128. The resource providing nodes 120, 122, 124, 126, 128 communicate with resource providers 130, 132, 134, 136 and among themselves to maintain local databases of resource availability including quality of service specifications, and cost. In the preferred embodiment, the resource availability  
20 data at the resource providing nodes 120, 122, 124, 126, 128 include different levels of quality of service for different costs. Preferably, RPNs 120, 122, 124, 126, 128 also make decisions about resource allocation on servers. These  
25 decisions may involve, but are not limited to, duplication of data or application resources, or moving data, application or license resource from one server to another.

#### **Resource Allocation Method**

FIG. 2 provides a flow diagram describing a method  
30 200 executed by the resource providing nodes 120, 122, 124, 126, 128. As is known to persons of ordinary skill in the

art, a dataflow diagram is a graph whose nodes are processes and whose arcs are dataflows. See *Object Oriented Modeling and Design*, Rumbaugh, J., Prentice Hall, Inc. (1991), Chapter 1. In step 210, the method 200 receives resource requests  
5 110, 112, 114, 116. In step 220, the method 200 receives resource offers 130, 132, 134, 136 from resource providers. In step 230, the method 200 combines the resources offers 130, 132, 134, 136 to form new resource offer combinations. In step 240, the method 200 searches for relations between  
10 the resource requests 110, 112, 114, 116 and the resource offers 110, 112, 114, 116 as well as the new resource offer combinations.

In step 250, the method 200 evaluates the relations between the resource requests 110, 112, 114, 116 and the  
15 resource offers 110, 112, 114, 116 as well as the new resource offer combinations. In step 260, the method selects the relations that are optimal with respect to the evaluation and allocates the selected relation's resource offer to its corresponding resource request.

20

### **Technology Graph**

Execution of step 230 yields a technology graph 235, which is a multigraph representation of the processes executed by the resource providing nodes 120, 122, 124, 126  
25 and 128 to form combinations of resources to satisfy the resource requests 110, 112, 114, 116. FIG. 3 displays an exemplary technology graph. As is known to persons of ordinary skill in the art, a multigraph is a pair  $(V, E)$  where  $V$  is a set of vertices,  $E$  is a set of hyperedges, and  $E$  is a subset of  $P(V)$ , the power set of  $V$ . See *Graph Theory*, Bela  
30 Bollobas, Springer-Verlag, New York, 1979, ("Graph Theory") Chapter 1. The power set of  $V$  is the set of subsets of  $V$ .

See *Introduction to Discrete Structures*, Preparata and Yeh, Addison-Wesley Publishing Company, Inc. (1973) ("Introduction to Discrete Structures"), pg 216.

5 In the technology graph  $(V, E)$ , each vertex  $v$  of the set of vertices  $V$  represents a resource. More formally, there exists a one-to-one correspondence between the set of resources and the set of vertices  $V$  in the technology graph  $(V, E)$ . A function denoted by  $g: O \rightarrow V$  from the set of resources  $O$  representing the resources to the set of vertices  
10  $V$  in the corresponding multigraph  $(V, E)$  assigns the vertex  $v$  to a resource  $o$  ( $g(o) = v$ ).

In the technology graph  $(V, E)$ , each hyperedge  $e$  of the set of hyperedges  $E$  represents a transformation as shown by FIG. 3. The outputs of the hyperedge  $e$  are defined as the  
15 intermediate resources 310 or the finished resources 315 produced by execution of the transformation represented by the hyperedge  $e$ . The outputs of the hyperedge  $e$  also include the waste products of the transformation. The inputs of the hyperedge  $e$  represent the complementary resources used in the  
20 production of the resources of the hyperedge. Complementary resources are used jointly to produce other resources.

The technology graph includes resources 305, intermediate resources 310, finished resources 315, and machines 320. Machines 320 are resources which perform  
25 ordered sequences of transformations on an input bundle of resources to produce an output bundle of resources. Accordingly, resources 310 are produced when machines 320 execute their transformations on an input bundle of resources. A machine 320 which mediates transformations is  
30 represented in the technology graph  $H = (V, E)$  as an input to a hyperedge  $e$ . In an alternate embodiment, a machine 320 which mediates transformations is represented as an object

which acts on the hyperedge  $e$  to execute the transformation. Finished resources 315 are the end products which are produced for the client making a resource request 110, 112, 114, 116.

5           The objects and transformations among the resources in the technology graph  $H = (V, E)$  constitute a generative grammar. As is known by persons of ordinary skill in the art, context-free grammars represent transformations or productions on symbol strings. Each production specifies a  
10 substitute symbol string for a given symbol string. The technology graph  $H = (V, E)$  extends the principles of context-free grammars from symbol strings and transformations among symbol strings to resources and transformations among resources. The expressiveness of the technology graph  $H =$   
15  $(V, E)$  is higher than that of context-free grammars as hypergraphs can represent multidimensional relationships directly. The technology graph  $H = (V, E)$  also expresses a context sensitive grammar.

Each transformation in the technology graph  $H = (V,$   
20  $E)$  may specify a substitute hypergraph for a given hypergraph. Accordingly if a subgraph within a hypergraph matches a given hypergraph in a transformation, the subgraph is removed and replaced by the substitute hypergraph. The resulting hypergraph is derived from the original hypergraph.

FIG. 4 provides a dataflow diagram 400 representing  
25 an overview of a method for synthesizing the technology graph. In step 410, the technology graph synthesis method performs the initialization step. The technology graph synthesis method initializes the set of vertices  $V$  of the technology graph  $H = (V, E)$  to a *founder set* of resources.  
30 The *founder set* contains the most primitive resources. Thus, the *founder set* could represent renewable resources. The

*founder set* can have from zero to a finite number of resources. The method also initializes a set of transformations,  $T$ , with a finite number of predetermined transformations in step 410. Finally, the method initializes  
5 an iterate identifier,  $i$ , to 0 in step 410.

In step 415, the method determines whether the iterate identifier is less than a maximum iterate value,  $I$ . If the iterate identifier is not less than the maximum iterate value,  $I$ , the method terminates at step 430. If the  
10 iterate identifier is less than the maximum iterate value,  $I$ , then control proceeds to step 420.

In step 420, the technology graph synthesis method applies the set of transformations,  $T$ , to the set of vertices  $V$ . In the first iteration of the loop of the flow diagram of  
15 FIG. 4, step 420 applies the set of transformations,  $T$ , to the resources in the *founder set*. First, step 420 applies each transformation in the set of transformations,  $T$ , to each resource in the *founder set*. Next, step 420 applies each transformation in the set of transformations,  $T$ , to all  
20 pairs of resources in the *founder set*. Step 420 similarly continues by applying each transformation in the set of transformations,  $T$ , to each higher order subset of resources in the *founder set*. Execution of step 420 in iteration,  $i$ , yields the  $i$  *th* technology adjacent possible set of  
25 resources. Execution of step 420 in iteration,  $i$ , also yields a modified technology graph  $H = (V, E)$ . The modified technology graph  $H = (V, E)$  contains additional vertices corresponding to the  $i$  *th* technology adjacent possible set of resources and additional hyperedges  $e$  corresponding to the  
30 transformations applied in the  $i$  *th* iteration of the loop of the flow graph of FIG. 4.

In one embodiment, the method maintains all vertices created by execution of step 420 in the technology graph  $H = (V, E)$ . In an alternate embodiment, step 425 prunes all vertices representing duplicate elements of the  
5 *ith technology adjacent possible* set of resources from the technology graph  $H = (V, E)$ . Accordingly, in the first embodiment of step 425, every resource constructed at each iteration of the method is kept in the technology graph  $H = (V, E)$ . Execution of the technology graph synthesis method  
10 400 using the first embodiment of step 425 produces a *full* technology graph  $H = (V, E)$ . In the alternate embodiment, only resources which were not elements in the *founder set* and which were not created in previous iterations of the loop of the flow diagram of FIG. 4 are added to the technology graph  
15  $H = (V, E)$ . Execution of the technology graph synthesis method 400 using the alternate embodiment with the pruning of step 425 produces a *minimal* technology graph  $H = (V, E)$ . After execution of step 425, control returns to step 415.

In subsequent iterations of the loop of the flow graph of FIG. 4, step 420 applies the set of transformations,  
20  $T$ , to the resources in the set of vertices  $V$  of the technology graph  $H = (V, E)$  produced by the execution of the previous iteration of the loop.

The set of transformations  $T$  can be held fixed  
25 throughout the execution of the technology graph synthesis method 400. Alternatively, new transformations could be added to the set of transformations and old transformations could be removed. For example, resources representing machines could also be included in the *founder set* of  
30 resources. Next, the set of transformations  $T$  could be applied to the resources representing machines just as they are applied to the other resources in the technology graph  $H$

$= (V, E)$ . Consequently, the set of transformations  $T$  could be limited to the transformations which are mediated by those machine resources represented by vertices of the technology graph  $H = (V, E)$ .

5

### Technology Graph Applications

The paths in the technology graph  $H = (V, E)$  which begin at vertices corresponding to resources in the *founder set* and end at vertices corresponding to finished goods represent the *processes* for producing the finished resources from the resources in the founder set. A path  $P_i$  of a hypergraph  $H = (V, E)$  is defined as an alternating sequence of vertices and edges  $v_{i1}, e_{i1}, v_{i2}, e_{i2}, v_{i3}, e_{i3}, v_{i4}, e_{i4}, \dots$  such that every pair of consecutive vertices in  $P_i$  are connected by the hyperedge  $e$  appearing between them along  $P_i$ . As previously discussed, the vertices of the technology graph represent renewable resources, intermediate resources and finished resources and the hyperedges of the technology graph represent transformations. Accordingly, a path  $P_i$  in the technology graph  $H = (V, E)$  from a founder set to a finished resource identifies the renewable resources, the intermediate resources, the finished resources, the transformations and the machines mediating the transformations of the process. Thus, a *process* is also referred to as a *construction pathway*.

25

The technology graph  $H = (V, E)$  also contains information defining a first *robust constructability* measure of a terminal resource representing a finished resource. The first *robust constructability* measure for a terminal resource is defined as the number of *processes* or *construction pathways* ending at the terminal resource. *Process redundancy* for a terminal resource exists when the number of *processes*

30

or *construction pathways* in a technology graph exceeds one. Failures such as an interruption in the supply of a renewable resource or the failure of a machine cause *blocks* along *construction pathways*. Greater numbers of *processes* or

5 *construction pathways* to a terminal resource indicate a greater probability that a failure causing *blocks* can be overcome by following an alternate *construction pathway* to avoid the *blocks*. Accordingly, higher values of the first

10 *robust constructability* measure for a terminal resource indicate higher levels of reliability for the *processes* which produce the finished resource represented by the terminal resource. Further, the technology graph extends the traditional notion of the *makespan*.

The technology graph  $H = (V, E)$  also contains

15 information defining a second *robust constructability* measure of a terminal resource representing a finished resource. The second *robust constructability* measure for a terminal resource is defined as the rate at which the number of

20 *processes* or *construction pathways* ending at the terminal resource increases with the *makespan* of the process. For example, suppose a terminal resource can be constructed with a *makespan* of  $N$  time steps with no *process redundancy*. Since there is no *process redundancy*, a *block* along the only

25 *construction pathway* will prevent production of the terminal resource until the cause of the *block* is corrected. The relaxation of the required *makespan* to  $N + M$  time steps will increase the number of *construction pathways* ending at the terminal resource. Accordingly, failures causing *blocks* can be overcome by following an alternate *construction pathway* to

30 the terminal resource. In other words, while the minimum possible *makespan* increased by  $M$  time steps, the resulting greater numbers of *processes* or *construction pathways* to the

terminal resource led to greater reliability. Thus, the present invention extends the notion of a *makespan* to include the concept of *robust constructability*.

The additional *robust constructability* measures  
5 involve vertices which exist within the *construction pathways* of two or more terminal resources. These resources represented by these vertices are called *poly-functional intermediate resources* because two or more terminal resources can be constructed from them. For example, consider two  
10 terminal resources representing a house and a house with a chimney. The *poly-functional intermediate resources* are the resources represented by vertices which exist within a *construction pathway* of the house and within a *construction pathway* of the house with the chimney. Thus, if a consumer  
15 requests a chimney in a house after the house has been constructed without a chimney, the chimney can be added to the house by backtracking along the *construction pathway* of the house to a *poly-functional intermediate resource* and proceeding from the *poly-functional intermediate resource*  
20 along a *construction pathway* of the house with a chimney.

FIG. 5 provides a flow diagram 500 for locating and selecting *poly-functional intermediate resources* for a set of terminal resources 501 having a cardinality greater than or equal to two. In step 504, the method determines the  
25 vertices which exist within the *construction pathways* of each terminal resource in the set of terminal resources 501 in the technology graph  $H = (V, E)$ . Execution of step 504 yields a set of vertices 505 for each terminal resource in the set of terminal resources 501. Accordingly, the number of sets of  
30 vertices 505 resulting from execution of step 504 is equal to the cardinality of the set of terminal resources 501. In step 506, the method performs the intersection operation on

the sets of vertices 505. Execution of step 506 yields the vertices which exist within the *construction pathways* of every terminal resource in the set of terminal resources 501. In other words, execution of step 506 yields the *poly-functional intermediate resources* 507 of the set of terminal resources 501.

In step 508, the method performs a selection operation on the *poly-functional intermediate resources* 507. Preferably, step 508 selects the *poly-functional intermediate resource* 707 with the smallest *fractional construction pathway distance*. The *fractional construction pathway distance* of a given *poly-functional intermediate resource* is defined as the ratio of two numbers. The numerator of the ratio is the sum of the smallest distances from the given *poly-functional intermediate resource* to each terminal resource in the set of terminal resources 501. The denominator of the ratio is the sum of the numerator and the sum of the smallest distances from each resource in the *founder set* to the given *poly-functional intermediate resource*. The distance between two vertices along a *construction pathway* in the technology graph  $H = (V, E)$  is defined as the number of hyperedges  $e$  on the *construction pathway* between the two vertices. The smallest distance between two vertices in the technology graph  $H = (V, E)$  is the number of hyperedges  $e$  on the shortest *construction pathway*.

Alternatively, step 508 considers the *process redundancy* in addition to the *fractional construction pathway distance* in the selection of the *poly-functional intermediate resource* 507. This alternative selection technique first locates the *poly-functional intermediate resource* 507 having the smallest *fractional construction pathway distance*. Next,

the alternative technique traverses the *construction pathways* from the *poly-functional intermediate resource* 507 having the smallest *fractional construction pathway distance* toward the *founder set* until it reaches a *poly-functional intermediate resource* 507 having a sufficiently high value of *process redundancy*.

The method of FIG. 5 for locating and selecting *poly-functional intermediate resources* for a set of terminal resources 501 can also be executed on different subsets of the power set of the set of terminal resources 501 to locate and select *poly-functional intermediate resources* for different subsets of the set of terminal resources.

As indicated by the preceding discussion, the present invention identifies and selects the *poly-functional resource* which leads to process redundancy to achieve reliability and adaptability. Specifically, the resource providing nodes should ensure that there is an adequate inventory of the selected *poly-functional resource* to enable it to adapt to failures and changes in the computing environment. Resource providing nodes use the technology graphs to determine how to satisfy the sub-requirements (e.g., in the "needs-a" property of a resource request) of a particular resource request.

Technology graphs are further explained in co-pending U.S. application number 09/345,441 filed July 1, 1999 and titled, "An Adaptive and Reliable System and Method for Operations Management", the contents of which are herein incorporated by reference.

### **Searching for Relations**

In one embodiment, the method of the present invention for allocating resources searches for relations

between the resource requests 110, 112, 114, 116 and the resource offers 130, 132, 134, 136 using an active search. An exemplary active search includes an ants based reinforcement learning algorithm as described in co-pending  
5 patent application No. 09/368,215, filed 8/04/1999, and titled, A Method and System for Controlling Network Traffic, the contents of which are herein incorporated by reference.

In another embodiment, the method of the present invention for allocating resources searches for relations  
10 between the resource requests 110, 112, 114, 116 and the resource providers 130, 132, 134, 136 uses a passive search. An exemplary passive search includes advertisements of the resource requests 110, 112, 114, 116 by clients and advertisements of the resource offers 130, 132, 134, 136 by  
15 resource providers.

#### **Market-Based Resource Allocation**

FIG. 6 displays a flow diagram of a method 600 for allocating resources using a market-based scheme which  
20 executes on a resource providing node 130, 132, 134, 136. In step 602, the market-based allocation method 600 receives resource requests 110, 112, 114, 116. In step 604, the method 600 receives bids for resource requests 110, 112, 114, 116 from other resource provider nodes 120, 122, 124, 126,  
25 128.

In the context of the market-based allocation method 600, the resource requests 110, 112, 114, 116 include a contract to pay a specified reward for a satisfaction of the request. The resource requests 110, 112, 114, 116  
30 further include a specified quality of service. In the preferred embodiment, the resource requests 110, 112, 114, 116 also includes a specified reward that varies with a

delivered quality of service in comparison with the specified quality of service. Preferably, the specified quality of service includes a time for the satisfaction of the resource request.

- 5           The contracted amount is paid in full only if the resource request 110, 112, 114, 116, is satisfied in accordance with its specified quality of service. Preferably, a portion of the contracted amount is paid to the resource providing node 120, 122, 124, 126, 128 that
- 10   satisfied the resource request 110, 112, 114, 116 if the satisfaction was outside the specified quality of service. This portion is determined as a function of the received quality of service. Preferably, less cash is released for resource requests 110, 112, 114, 116 that are satisfied long
- 15   after the specified time quality of service. Market-arbiter software calculates the cash reward earned by the satisfying resource providing node 120, 122, 124, 126, 128 and the amount owed by the client that transmitted the resource request 110, 112, 114, 116. These rewards and bills are
- 20   accumulated over time and sent out at a low frequency so as to impose only a negligible communication load on the system 100.

- The bids for resource requests 110, 112, 114, 116 from other resource providing nodes 120, 122, 124, 126 include a price that will be paid for the resource request.
- 25   Optionally, the bids for resource requests 110, 112, 114, 116 could also include an expiration time or a margin. In the preferred embodiment, the bids for resource requests 110, 112, 114, 116 includes a satisfaction profile defining a satisfaction of trading the resource request as a probability
- 30   density function of one or more parameters. Exemplary parameters include a quality of service. Satisfaction

profiles are described in co-pending U.S. application number 09/345,441, filed July 1, 1999, titled, "An Adaptive and Reliable System and Method for Operations Management", the contents of which are herein incorporated by reference.

5           In step 606, the method 600 computes an expected return for processing the resource request 110, 112, 114, 116. In step 608, the method 600 processes the resource request 110, 112, 114, 116 to optimize the expected return. Exemplary processing options include satisfying the resource  
10 request 110, 112, 114, 116, in step 610 or selling the resource request 110, 112, 114, 116 to another resource processing node 130, 132, 134, 136 in step 612.

          In step 614, the method 600 transmits bids for the resource requests 110, 112, 114, 116 to other resource  
15 provider nodes 130, 132, 134, 136.

          Each resource providing node 130, 132, 134, 136 acts autonomously to optimize the value of some function combining its own expected return and that of some (zero or more) selected neighbors (not necessarily immediate  
20 topological neighbors) as explained more fully below. The expected return is based on its earnings from satisfying or selling resources requests 110, 112, 114, 116. Optionally, resource providing nodes 130, 132, 134, 136 learn to optimize their expected return using reinforcement learning. An  
25 exemplary reinforcement learning technique is Q-learning.

          Resource providing nodes 120, 122, 124, 126, 128 receive feedback about their performance. This feedback is called a *reward*. However, in the reinforcement learning framework of the present invention, a resource providing  
30 nodes 120, 122, 124, 126, 128 does not merely act to optimize its immediate reward. Instead, it acts to optimize its *return*. In the preferred embodiment, the *return* includes an

expected future reward that is discounted to present value. As mentioned earlier, reward is based on "earnings" in a resource market called the market-based reward framework.

When reinforcement learning is used to adjust the behavior of resource providing nodes 120, 122, 124, 126, 128, instantaneous rewards are based on the actual cash profit of the resource providing nodes 120, 122, 124, 126, 128 and optionally, the cash profit of neighboring resource providing nodes 120, 122, 124, 126, 128 (not necessarily topological neighbors) over some short past time period. Optionally, in order to prevent resource providing nodes 120, 122, 124, 126, 128 from charging arbitrary prices in monopoly situations, excess profit can be removed (taxed) from those resource providing nodes 120, 122, 124, 126, 128 whose long-term discounted expected reward exceeds a predefined target.

Each resource providing node 120, 122, 124, 126, 128 communicates "bids" that specify how much it will pay for resource requests 110, 112, 114, 116 having a particular specified quality of service, and a specified reward to other resource providing nodes 120, 122, 124, 126, 128. Preferably, each resource providing nodes 120, 122, 124, 126, 128 communicates the "bids" to its topologically neighboring agents. Bids may also have an expiration time. Bids stand until they expire or until the resource providing nodes 120, 122, 124, 126, 128 where a bid is held receives a message canceling and/or replacing the bid. Optionally, other quality of service parameters corresponding to the quality of service requirements of resource requests 110, 112, 114, 116 are included in the bids.

Resource requests 110, 112, 114, 116 that are received by a resource providing node 120, 122, 124, 126, 128 (either from an application program at the resource providing

nodes 120, 122, 124, 126, 128, or from another resource providing nodes 120, 122, 124, 126, 128) that do not conform to the parameters of an existing bid (e.g., insufficient contract value or too many in a given time period) do not  
5 require payment. Instead, these resource requests 110, 112, 114, 116 are owned by the resource providing nodes 120, 122, 124, 126, 128, and may be sold.

A market-based allocation method for data routing is explained in co-pending international patent application,  
10 having attorney docket number 9392-025-228, filed January 28, 2000, and titled, "A Method and System for Routing Control in Communication Networks and for System Control", the contents of which are herein incorporated by reference.

## 15 Locally-cooperative local reinforcement learning

Having all resource providing nodes 120, 122, 124, 126, 128 attempt to optimize their local expected return will not always result in the discovery of the globally optimum configuration as explained in "At Home in the Universe" by  
20 Stuart Kauffman, Oxford University Press, Chapter 11 in the context of an NK fitness landscape, the contents of which are herein incorporated by reference. This result occurs because actions taken by one resource providing node 120, 122, 124,  
25 126, 128 affects its state and possibly changes the context of the reward for its neighboring resource providing nodes 120, 122, 124, 126, 128.

Accordingly, in the preferred embodiment the present invention utilizes combinations of the following three semi-local strategies:  
30

**patches** In this technique, resource providing nodes 120, 122, 124, 126, 128 are partitioned into disjoint subsets called patches. The patches may or may not be topologically contiguous. Within a patch, the actions of resource providing nodes 120, 122, 124, 126, 128 are coordinated to maximize the aggregate figure of merit for the entire patch. The size and location of patches are parameters for this strategy.

**p** A neighborhood is defined for a resource providing nodes 120, 122, 124, 126, 128 such that when a decision is made there, figures of merit at the current node and at a proportion  $p$  of neighboring nodes are taken into account. A neighborhood need not consist of the immediate topological neighbors of the resource providing nodes 120, 122, 124, 126, 128.

**tau** Only a fraction (called  $\tau$ ) of the resource providing nodes 120, 122, 124, 126, 128 make decisions that change the portions of their state that affect the reward of other resource providing nodes 120, 122, 124, 126, 128 at the same time.

FIG. 7 provides a flow diagram 700 for determining optimal values of parameters of methods performing resource allocation and load balancing. In step 710, the present invention defines a global performance measure for the network. In step 720, the present invention defines an optimization algorithm having at least one parameter. Exemplary parameters include the size and location of patches, the neighborhood,  $p$  where the expected reward are

considered in making a decision and the fraction,  $\tau$ , of the agents that change portions of their state that affect the reward of other agents. In step 730, the method 700 constructs a landscape representation for values of the parameters and their associated global performance measure. In step 740, the method optimizes over the landscape to produce optimal values for the parameters.

In the preferred embodiment, the present invention uses either patches or  $p$  or both to define a modified reward and hence, a return, for a resource providing nodes 120, 122, 124, 126, 128 in the resource allocation problem. As explained earlier, the figure of merit for a resource providing nodes 120, 122, 124, 126, 128 is its earnings in the market-based framework. Optionally, the present invention uses the  $\tau$  strategy either alone, or in conjunction with  $p$  and "patches" to limit the opportunities resource providing nodes 120, 122, 124, 126, 128 have for making decisions that affect the return of other resource providing nodes 120, 122, 124, 126, 128. For example, the reward for a resource providing node 120, 122, 124, 126, 128 is the aggregate earnings for a region of resource providing nodes 120, 122, 124, 126, 128 (a patch) and the bids and for only a fraction  $\tau$  of resource providing nodes 120, 122, 124, 126, 128 change at the same time.

Preferably, the parameters for these strategies (the fraction  $p$ , the fraction  $\tau$  and the number and membership of patches) are global in nature. In other words, the values of these parameters are the same for all resource providing nodes 120, 122, 124, 126, 128. Alternatively, the values of the parameters may vary among the resource providing nodes 120, 122, 124, 126, 128.

Preferably, the present invention sets these parameters as follows:

First, a global performance measure is defined. Preferably, the global performance measure is the specified  
5 quality of service in relation to the quality of service of the satisfied resource request 110, 112, 114, 116. Second, the algorithm has an outer loop that varies these parameters in order to maximize the global performance measure in accordance with techniques for searching landscapes as also  
10 described in the co-pending international patent application having attorney-docket number 9392-016-228, titled, "A System and Method for the Analysis and Prediction of Economic Markets", filed December 22, 1999 at the U.S. receiving office, the contents of which are herein incorporated by  
15 reference.

Preferably, each value of the global parameters governing  $p$ , patches,  $\tau$ , and reinforcement learning features, defines a point in the global parameter space. With respect to this point, the method for allocating  
20 resources of the present invention achieves a given global fitness. The distribution of global fitness values over the global parameter space constitutes a "fitness landscape" for the entire bandwidth-agent system. Such landscapes typically have many peaks of high fitness, and statistical features  
25 such as correlation lengths and other features as described in co-pending international patent application number PCT/US99/19916, titled, "A Method for Optimal Search on a Technology Landscape", the contents of which are herein incorporated by reference. In the preferred embodiment,  
30 these features are used to optimize an evolutionary search in the global parameter space to achieve values of  $p$ , patches,  $\tau$ , and the internal parameters of the reinforcement

learning algorithm that optimize the learning performance of the resource allocation system in a stationary environment with respect to load and other use factor distribution. Preferably, the same search procedures are also used to

5 persistently tune the global parameters of the resource allocation system in a non-stationary environment with respect to load and other use factor distributions.

By tuning of the global parameters to optimize learning, the present invention is "self calibrating". In other

10 words, the invention includes an outer loop in its learning procedure to optimize learning itself, where co-evolutionary learning is in turn controlled by combinations of  $p$ , patches, and  $\tau$ , plus features of the reinforcement learning algorithm. The inclusion of features of fitness landscapes

15 aids optimal search in this outer loop for global parameter values that themselves optimize learning by the resource allocation system in stationary and non-stationary environments.

Use of  $p$ ,  $\tau$ , or patches aids adaptive search on rugged

20 landscapes because, each by itself, causes the evolving system to ignore some of the constraints some of the time. Judicious balancing of ignoring some of the constraints some of the time with search over the landscape optimizes the balance between "exploitation" and "exploration". In

25 particular, without the capacity to ignore some of the constraints some of the time, adaptive systems tend to become trapped on local, very sub-optimal peaks. The capacity to ignore some of the constraints some of the time allows the total adapting system to escape badly sub-optimal peaks on the fitness landscape and thereby, enables further searching.

30 In the preferred embodiment, the present invention tunes  $p$ ,  $\tau$ , or patches either alone or in conjunction with one

another to find the proper balance between stubborn exploitation hill climbing and wider exploration search.

The optimal character of either  $\tau$  alone or patches alone, is such that the total adaptive system is poised  
5 slightly in the ordered regime, near a phase transition between order and chaos. See e.g. "At Home in the Universe" by Kauffman, Chapters 1,4, 5 and 11, the contents of which are herein incorporated by reference and "The Origins of Order, Stuart Kauffman, Oxford University Press, 1993,  
10 Chapters 5 and 6, the contents of which are herein incorporated by reference. For the  $p$  parameter alone, the optimal value of  $p$  is not associated with a phase transition.

Without limitation, the embodiments of the present invention are described in the illustrative context of a  
15 solution using  $\tau$ ,  $p$ , and patches. However, it will be apparent to persons of ordinary skill in the art that other techniques that ignore some of the constraints some of the time could be used to embody the aspect of the present invention which includes defining an algorithm having one or more parameters, defining a global performance measure,  
20 constructing a landscape representation for values of the parameters and their associated global performance value, and optimizing over the landscape to determine optimal values for the parameters. Other exemplary techniques that ignore some of the constraints some of the time include simulated  
25 annealing, or optimization at a fixed temperature. In general, the present invention employs the union of any of these means to ignore some of the constraints some of the time together with reinforcement learning to achieve good problem optimization.

30 Further, there are local characteristics in the adapting system itself that can be used to test locally that the system is optimizing well. In particular, with patches

alone and  $\tau$  alone, the optimal values of these parameters for adaptation are associated with a power law distribution of small and large avalanches of changes in the system as changes introduced at one point to improve the system unleash  
5 a cascade of changes at nearby points in the system. The present invention includes the use of local diagnostics such as a power law distribution of avalanches of change, which are measured either in terms of the size of the avalanches, or in terms of the duration of persistent changes at any  
10 single site in the network.

The present invention's use of any combination of the above strategies, together with reinforcement learning in any of its versions, give it an advantage over prior art resource allocation methods because these strategies address  
15 many problems that could arise including the following:

- slow convergence to optimal allocation patterns,
- oscillation of network load, and
- locally beneficial but globally harmful routing patterns.

Without limitation, the embodiments of the present  
20 invention have been described in the illustrative context of a method for allocating resources. However, it is apparent to persons of ordinary skill in the art that other contexts could be used to embody the aspect of the present invention which includes defining an algorithm having one or more  
25 parameters, defining a global performance measure, constructing a landscape representation for values of the parameters and their associated global performance value, and optimizing over the landscape to determine optimal values for the parameters.

30 For example, the present invention could be used for operations management as explained in co-pending U.S. patent application No. 09/345,441, titled, "An Adaptive and

Reliable System and Method for Operations management" and filed on July 1, 1999, the contents of which are herein incorporated by reference. That patent application describes a model of an enterprise in its competitive environment,

5 based on technology graphs that support a nodes and flow model of an organization, plus a management structure. The present invention, using agents to represent resources and operations in the enterprise model, coupled to reinforcement learning,  $p$ , patches and  $\tau$ , is used advantageously to

10 create a model of a learning organization that learns how to adapt well in its local environment. By use of the outer loop described above, good global parameter values for  $p$ , patches,  $\tau$ , and the reinforcement learning algorithm are discovered. In turn, these values are used to help create

15 homologous action patterns in the real organization. For example, the homologous action patterns can be created by tuning the partitioning of the organization into patches, by tuning how decisions at one point in the real organization are taken with respect to a prospective benefit of a fraction

20  $p$  of the other points in the organization affected by the first point, and by tuning what fraction,  $\tau$ , of points in the organization should try operational and other experiments to improve performance.

In addition, the distribution of contract values and rewards in the reinforcement algorithm can be used to

25 help find good incentive structures to mediate behavior by human agents in the real organization to achieve the overall adaptive and agile performance of the real organization.

In addition to the use of the invention to find good global parameters to instantiate in the real

30 organization, the same invention can be used to find good global parameter values to utilize in the model of the

organization itself to use that model as a decision support tool, teaching tool, etc.

Further, the present invention is also applicable to portfolio management, risk management, scheduling and  
5 routing problems, logistic problems, supply chain problems and other practical problems characterized by many interacting factors.

FIG. 8 discloses a representative computing system 810 in conjunction with which the embodiments of the present  
10 invention may be implemented and executed. Computing system 810 may be a personal computer, workstation, or a larger system such as a minicomputer. However, one skilled in the art of computer systems will understand that the present invention is not limited to a particular class or model of  
15 computer.

As shown in FIG. 9, representative computing system 810 includes a central processing unit (CPU) 812, a memory unit 814, one or more storage devices 816, an input device 818, an output device 820, and communication interface 822.  
20 A system bus 824 is provided for communications between these elements. Computing system 810 may additionally function through use of an operating system such as Windows, DOS, or UNIX. However, one skilled in the art of computing systems will understand that the present invention is not limited to a particular configuration or operating system.  
25

Storage devices 816 may illustratively include one or more floppy or hard disk drives, CD-ROMs, DVDs, or tapes. Input device 818 comprises a keyboard, mouse, microphone, or other similar device. Output device 820 is a computer  
30 monitor or any other known computer output device. Communication interface 822 may be a modem, a network

interface, or other connection to external electronic devices, such as a serial or parallel port

While the above invention has been described with  
5 reference to certain preferred embodiments, the scope of the  
present invention is not limited to these embodiments. One  
skill in the art may find variations of these preferred  
embodiments which, nevertheless, fall within the spirit of  
the present invention, whose scope is defined by the claims  
10 set forth below.

15

20

25

30

## Claims

1. A method for resource allocation comprising  
5 the steps of:  
receiving a plurality of resource requests and a  
plurality of resource offers, said resource requests and said  
resource offers having one or more properties; and  
determining at least one relation between at least  
10 one of said resource requests and at least one of said  
resource offers to identify at least one matching said  
resource offer for said resource request; and  
allocating said at least one resource request with  
said at least one matching resource offer.
- 15 2. A method for allocating resources as in claim 1  
wherein said determining at least one relation step comprises  
the step of:  
searching for said at least one relation between  
20 said at least one of said resource requests and said at least  
one of said resource offers.
3. A method for allocating resources as in claim 2  
wherein said searching for at least one relation step further  
25 comprises the step of:  
identifying matching ones of said one or more  
properties of said resource offers with said one or more  
properties of said resource requests.
- 30 4. A method for allocating resources as in claim 3  
wherein said determining at least one relation step further  
comprises the step of:

evaluating said at least one relation by  
determining how well said one or more properties of said at  
least one resource offer match with said one or more  
properties of said at least one resource request; and  
5 selecting said at least one relation that is  
optimal with respect to said evaluation.

5. A method for resource allocation as in claim 1  
wherein said one or more properties comprise at least one  
10 identity.

6. A method for resource allocation as in claim 1  
wherein said one or more properties comprise at least one  
ability.

15 7. A method for resource allocation as in claim 1  
wherein said one or more properties comprise at least one  
requirement.

20 8. A method for resource allocation as in claim 1  
wherein said one or more properties comprise at least one  
quality of service.

25 9. A method for resource allocation as in claim 1  
wherein said plurality of resource offers comprise data  
resources.

30 10. A method for resource allocation as in claim 1  
wherein said plurality of resource offers comprise software  
resources.

11. A method for resource allocation as in claim 1 wherein said plurality of resource offers comprise computational resources.

5 12. A method for resource allocation as in claim 1 wherein said plurality of resource offers comprise communication resources.

10 13. A method for resource allocation as in claim 1 wherein said determining at least one relation step comprises the step of determining at least one complement relation between said at least one resource request and said at least one resource offer.

15 14. A method for resource allocation as in claim 1 wherein said determining at least one relation step comprises the step of determining at least one substitute relation between said at least one resource request and said at least one resource offer.

20 15. A method for resource allocation as in claim 1 further comprising the step of constructing a graph representing said at least one resource request, said at least one resource offer and said at least one relation.

25 16. A method for resource allocation as in claim 15 wherein said constructing a graph step comprises the steps of:

30 creating one or more vertices representing said at least one resource offer and said at least one resource request;

creating at least one edge representing said at least one relation, said at least one edge having terminals corresponding to said at least one resource offer and said at least one resource request having said at least one relation.

5

17. Computer executable software code stored on a computer readable medium, the code for resource allocation the code comprising:

code to receive a plurality of resource requests  
10 and a plurality of resource offers, said resource requests and said resource offers having one or more properties; and

code to determine at least one relation between at least one of said resource requests and at least one of said resource offers to identify at least one matching said  
15 resource offer for said resource request; and

code to allocate said at least one resource request with said at least one matching resource offer.

18. A programmed component for resource allocation  
20 comprising at least one memory having at least one region storing computer executable program code and at least one processor for executing the program code stored in said memory, wherein the program code comprises:

code to receive a plurality of resource requests  
25 and a plurality of resource offers, said resource requests and said resource offers having one or more properties; and

code to determine at least one relation between at least one of said resource requests and at least one of said resource offers to identify at least one matching said resource offer for said resource request; and

30 code to allocate said at least one resource request with said at least one matching resource offer.

19. A method for allocating a plurality of resources comprising the steps of:

receiving information for at least one resource  
5 request;

computing an expected return for processing said resource request from said information; and

processing said at least one resource request to optimize said expected return.

10

20. A method for allocating a plurality of resources as in claim 19 wherein said information for said at least one resource request comprises a contract to pay a specified reward for a satisfaction of said at least one  
15 resource request.

15

21. A method for allocating a plurality of resources as in claim 20 wherein said information for said at least one resource request further comprises a specified  
20 quality of service.

20

22. A method for allocating a plurality of resources as in claim 21 wherein said specified reward varies with a satisfied quality of service in comparison with said specified quality of service.  
25

25

23. A method for allocating a plurality of resources as in claim 21 wherein said specified quality of service comprises a time for delivery of said satisfaction of said resource request.  
30

30

24. A method for allocating a plurality of resources as in claim 19 wherein said information for said at least one resource request comprises at least one bid specifying a price that will be paid for said at least one  
5 request.

25. A method for allocating a plurality of resources as in claim 24 wherein said at least one bid further comprises an expiration time.

10

26. A method for allocating a plurality of resources as in claim 24 wherein said at least one bid further comprises a margin.

15

27. A method for allocating a plurality of resources as in claim 24 wherein said at least one bid further comprises a satisfaction profile defining a satisfaction of trading said at least one request as a probability density function of at least one parameter.

20

28. A method for allocating a plurality of resources as in claim 27 wherein said at least one parameter of said probability density function comprises a quality of service.

25

29. A method for allocating a plurality of resources as in claim 19 wherein said expected return for said processing of said at least one resource request is an expected reward discounted to present value.

30

30. A method for allocating a plurality of resources as in claim 19 further comprising the step of

partitioning said plurality of resources into one or more patches.

31. A method for allocating a plurality of  
5 resources as in claim 30 wherein said processing said at least one resource request comprises the step of optimizing said expected return of said patch.

32. A method for allocating a plurality of  
10 resources as in claim 19 wherein said computing an expected return step comprises the step of:

selecting a portion  $p$  of said plurality of resources; and

15 computing said expected return of said selected portion  $p$  of said plurality of resources.

33. A method for allocating a plurality of  
resources as in claim 32 wherein said processing said at least one resource request comprises the step of optimizing  
20 said expected return of said selected portion  $p$  of said plurality of resources.

34. A method for allocating a plurality of  
resources as in claim 19 wherein said processing said at least one resource request comprises the step of satisfying  
25 said resource request.

35. A method for allocating a plurality of  
resources as in claim 19 wherein said processing said at least one resource request comprises the step of selling said  
30 resource request.

36. A method for allocating a plurality of resources as in claim 19 further comprising the step of transmitting at least one bid specifying a price that will be paid for said at least one resource request.

5

37. Computer executable software code stored on a computer readable medium, the code for allocating a plurality of resources, the code comprising:

code to receive information for at least one  
10 resource request;  
code to compute an expected return for processing said resource request from said information; and  
code to process said at least one resource request to optimize said expected return.

15

38. A programmed component for allocating a plurality of resources comprising at least one memory having at least one region storing computer executable program code and at least one processor for executing the program code  
20 stored in said memory, wherein the program code comprises:

code to receive information for at least one  
resource request;  
code to compute an expected return for processing said resource request from said information; and  
25 code to process said at least one resource request to optimize said expected return.

39. A method for allocating a plurality of resources comprising the steps of:

30 defining at least one algorithm having one or more parameters for allocating the resources;

defining at least one global performance measure of  
said at least one algorithm;

executing said algorithm for a plurality of  
different values of said one or more parameters to generate a  
5 corresponding plurality of values for said global performance  
measure;

constructing a fitness landscape from said values  
of said parameters and said corresponding values of said  
global performance measure; and

10 optimizing over said fitness landscape to generate  
optimal values for said at least one parameter.

40. A method for allocating a plurality of resources as  
in claim 39 wherein said defining at least one algorithm step  
15 comprises the steps of:

communicating information for at least one resource  
request;

computing an expected return for processing said  
resource request from said information; and

20 processing said at least one resource request to  
optimize said expected return.

41. A method for allocating a plurality of  
resources as in claim 40 wherein said at least one parameter  
25 comprises a proportion  $p$  of said plurality of resources.

42. A method for allocating a plurality of  
resources as in claim 41 wherein said computing an expected  
return step comprises the step of:

30 computing said expected return of said proportion  $p$   
of said one or more software agents.

43. A method for allocating a plurality of resources as in claim 40 wherein said at least one parameter comprises a size of one or more patches of said plurality of resources and a location of said patches.

5

44. A method for allocating a plurality of resources as in claim 43 wherein said processing said at least one resource request step comprises the step of: optimizing said expected return of said patch.

10

45. A method for allocating a plurality of resources as in claim 40 wherein said at least one parameter comprises a fraction,  $\tau$ , of said plurality of resources.

15

46. A method for allocating a plurality of resources as in claim 45 wherein only said fraction,  $\tau$ , of said plurality of resources communication information for said at least one resource request at one time.

20

47. Computer executable software code stored on a computer readable medium, the code allocating a plurality of resources, the code comprising:

code to define at least one algorithm having one or more parameters for allocating the resources;

25

code to define at least one global performance measure of said at least one algorithm;

code to execute said algorithm for a plurality of different values of said one or more parameters to generate a corresponding plurality of values for said global performance measure;

30

code to construct a fitness landscape from said values of said parameters and said corresponding values of said global performance measure; and

code to optimize over said fitness landscape to  
5 generate optimal values for said at least one parameter.

48. A programmed component for allocating a plurality of resources comprising at least one memory having at least one region storing computer executable program code  
10 and at least one processor for executing the program code stored in said memory, wherein the program code comprises:

code to define at least one algorithm having one or more parameters for allocating the resources;

code to define at least one global performance  
15 measure of said at least one algorithm;

code to execute said algorithm for a plurality of different values of said one or more parameters to generate a corresponding plurality of values for said global performance measure;

code to construct a fitness landscape from said  
20 values of said parameters and said corresponding values of said global performance measure; and

code to optimize over said fitness landscape to generate optimal values for said at least one parameter.

25

30

## Abstract

5           The present invention relates generally to a method  
and system for resource allocation. More particularly, the  
present invention allocates resources using technology  
graphs, passive and active searching, reinforcement learning,  
market driven decision making, reinforcement learning as well  
10 as  $p$ ,  $\tau$ , and patches techniques.

15

20

25

30

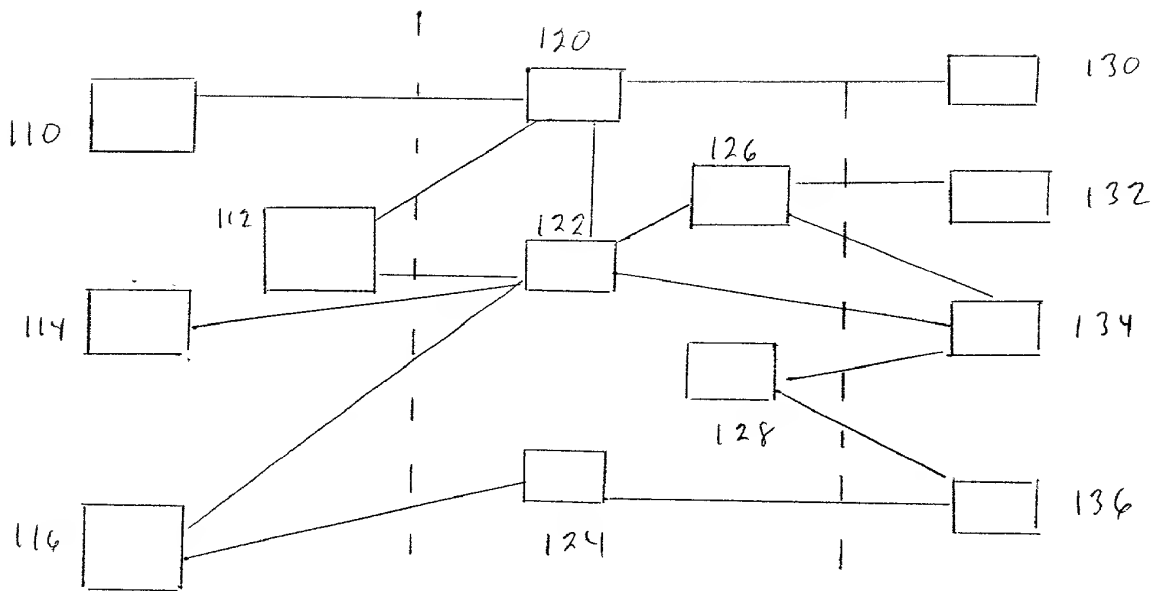


FIG. 1

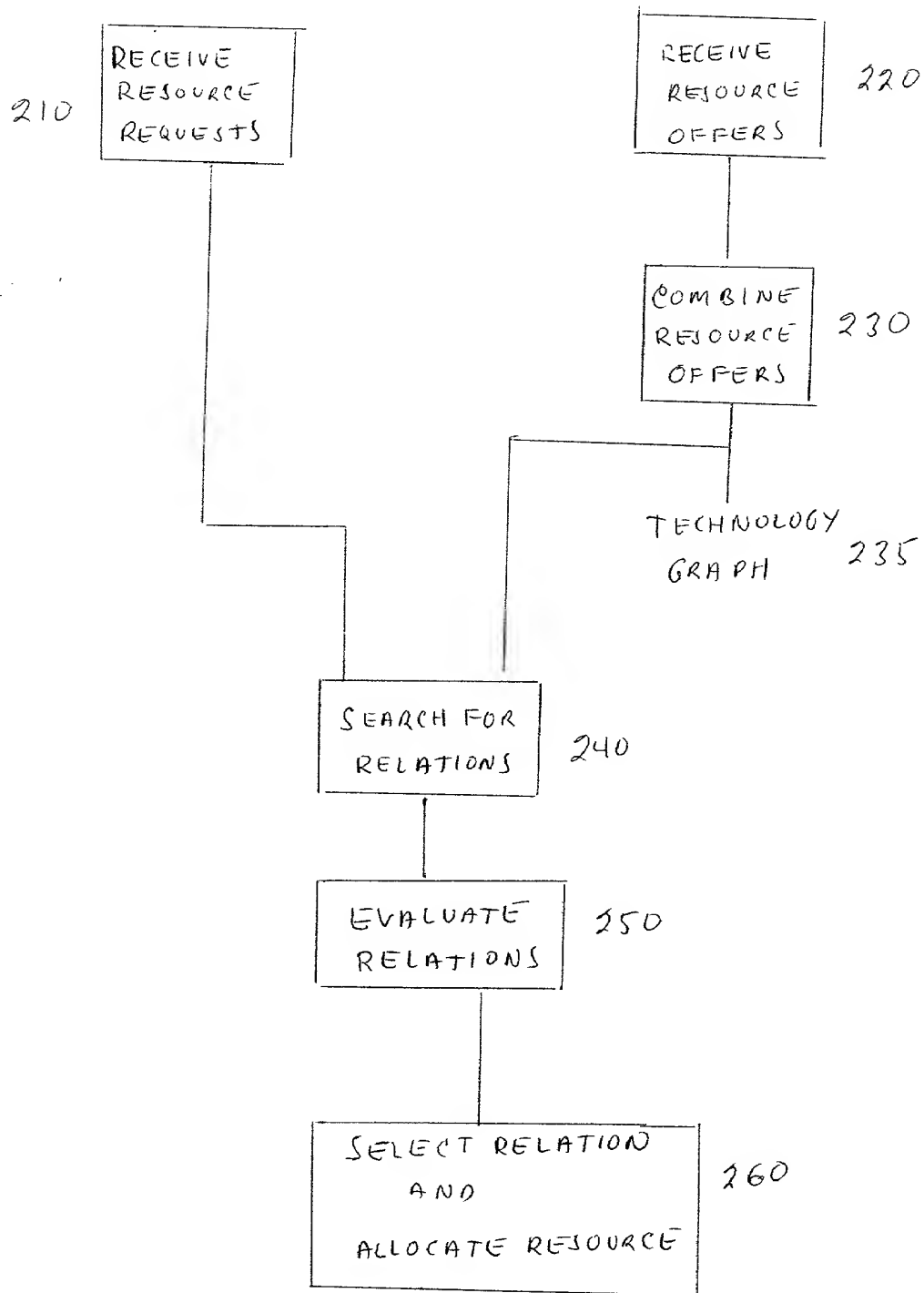


FIG. 2

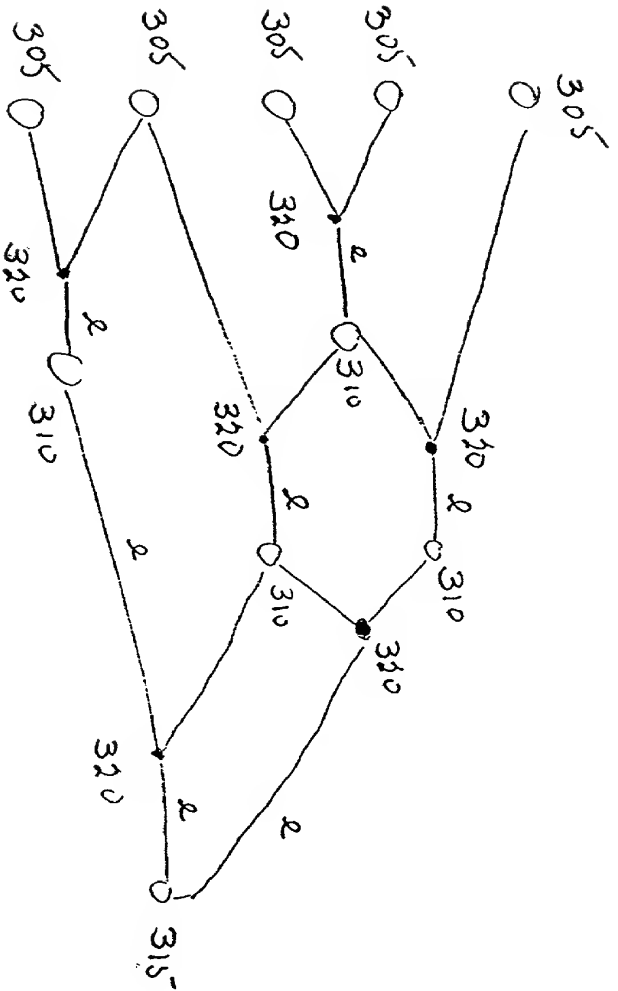


FIG. 3

410 Initialize: resources, transformations,  $i = 0$ ,  $H = (V, E)$

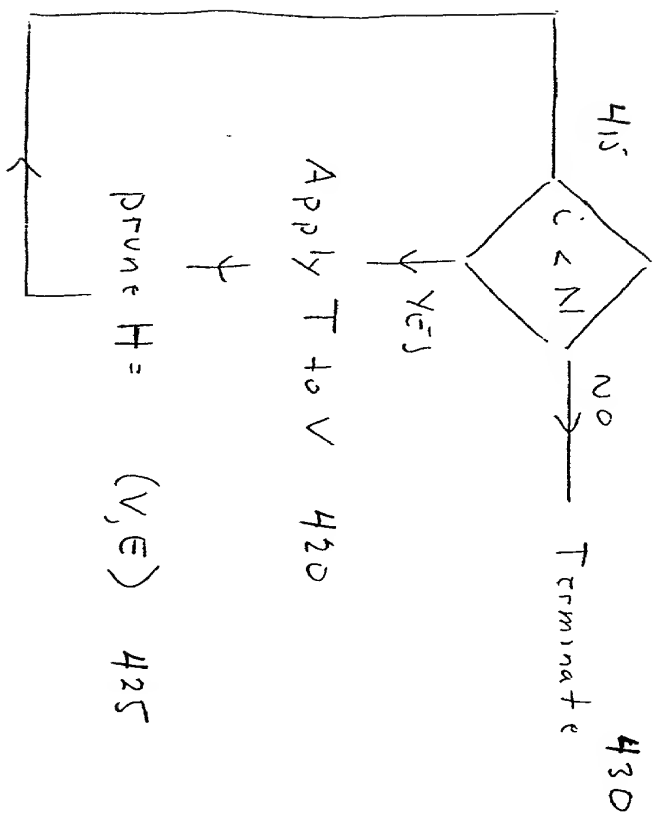


FIG. 4

400

501 {terminal resources} technology graph 503

X

X

determine vertices  
in construction  
pathway

504

{ { { { { ... { { { }

505

perform intersection

506

↓

polyfunctional objects 507

↓

[selection]

508

500

FIG. 5

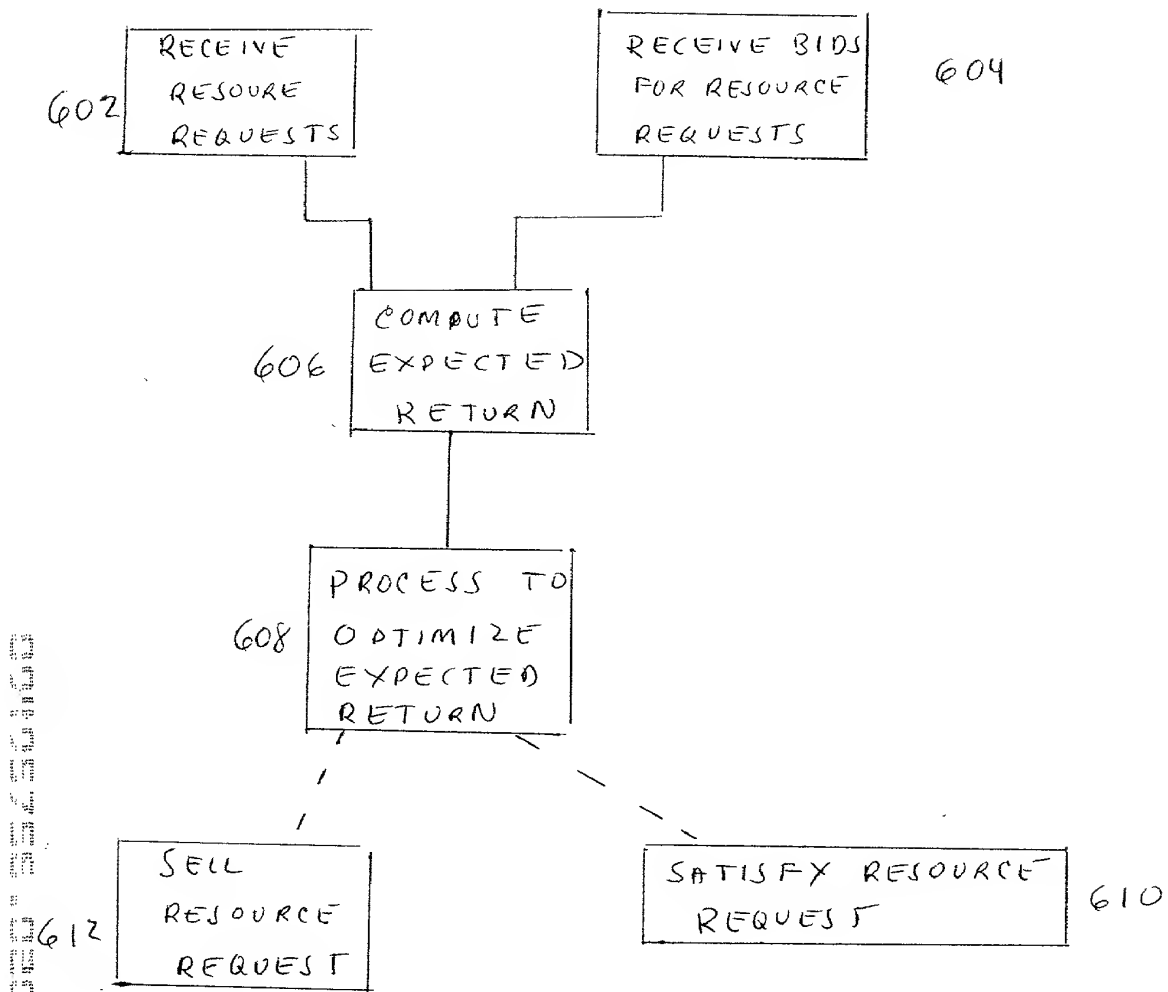


FIG. 6

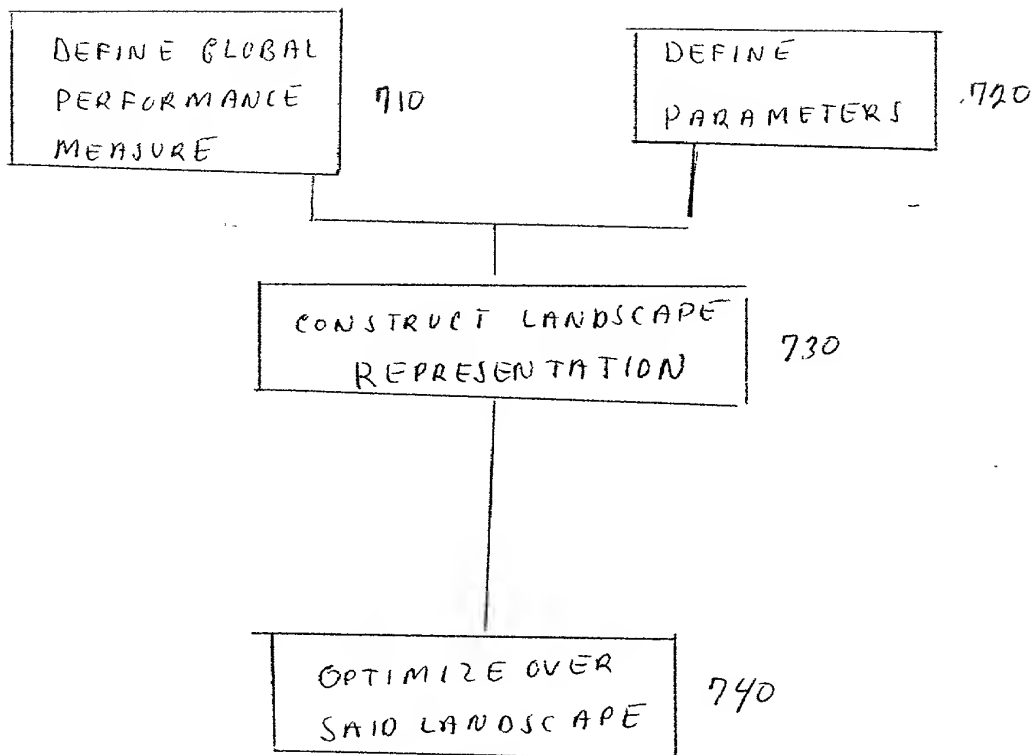


FIG. 7

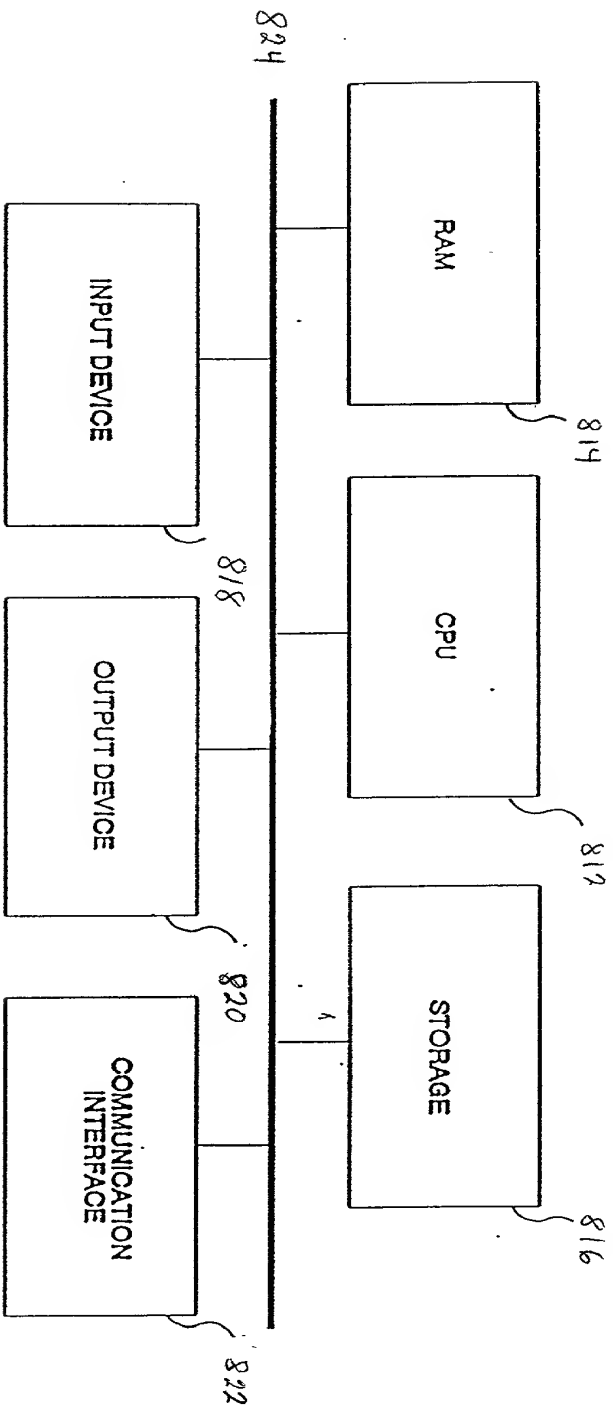


FIG. 8

# DECLARATION AND POWER OF ATTORNEY

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below at 201 et seq. underneath my name.

I believe I am the original, first and sole inventor if only one name is listed at 201 below, or an original, first and joint inventor if plural names are listed at 201 et seq. below, of the subject matter which is claimed and for which a patent is sought on the invention entitled

## A METHOD AND SYSTEM FOR ALLOCATION

and for which a patent application:

- ☒ is attached hereto and includes amendment(s) filed on \_\_\_\_\_ (if applicable)  
☐ was filed in the United States as Application No. \_\_\_\_\_ (for declaration not accompanying application)  
 with amendment(s) filed on \_\_\_\_\_ (if applicable)  
☐ was filed as PCT international Application No. \_\_\_\_\_ on \_\_\_\_\_ and was amended under PCT Article 19 on \_\_\_\_\_ (if applicable)

I hereby state that I have reviewed and understand the contents of the above identified application, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, § 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, § 119(a)-(d) of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

EARLIEST FOREIGN APPLICATION(S), IF ANY, FILED PRIOR TO THE FILING DATE OF THE APPLICATION			
APPLICATION NUMBER	COUNTRY	DATE OF FILING (day, month, year)	PRIORITY CLAIMED
			YES <input type="checkbox"/> NO <input type="checkbox"/>
			YES <input type="checkbox"/> NO <input type="checkbox"/>

I hereby claim the benefit under Title 35, United States Code, § 119(e) of any United States provisional application(s) listed below.

APPLICATION NUMBER	FILING DATE
60/118,174	February 1, 1999

I hereby claim the benefit under Title 35, United States Code, § 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code § 112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, § 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application:

APPLICATION SERIAL NO.	FILING DATE	STATUS		
		PATENTED	PENDING	ABANDONED

POWER OF ATTORNEY: As a named inventor, I hereby appoint S. Leslie Misrock (Reg. No. 18872), Harry C. Jones, III (Reg. No. 20280), Berj A. Terzian (Reg. No. 20060), David Weild, III (Reg. No. 21094), Jonathan A. Marshall (Reg. No. 24614), Barry D. Rein (Reg. No. 22411), Stanton T. Lawrence, III (Reg. No. 25736), Charles E. McKenney (Reg. No. 22795), Philip T. Shannon (Reg. No. 24278), Francis E. Morris (Reg. No. 24615), Charles E. Miller (Reg. No. 24576), Gidon D. Stern (Reg. No. 27469), John J. Lauter, Jr. (Reg. No. 27814), Brian M. Poissant (Reg. No. 28462), Brian D. Coggio (Reg. No. 27624), Rory J. Radding (Reg. No. 28749), Stephen J. Harbulak (Reg. No. 29166), Donald J. Goodell (Reg. No. 19766), James N. Palik (Reg. No. 25510), Thomas E. Friebe (Reg. No. 29258), Laura A. Coruzzi (Reg. No. 30742), Jennifer Gordon (Reg. No. 30753), Allan A. Fanucci (Reg. No. 30256), Geraldine F. Baldwin (Reg. No. 31232), Victor N. Balancia (Reg. No. 31231), Samuel B. Abrams (Reg. No. 30605), Steven I. Wallach (Reg. No. 35402), Marcia H. Sundeen (Reg. No. 30893), Paul J. Zegger (Reg. No. 33821), Edmond R. Bannon (Reg. No. 32110), Bruce J. Barker (Reg. No. 33291), Adriane M. Antler (Reg. No. 32605), Thomas G. Rowan (Reg. No. 34419), James G. Markey (Reg. No. 31636), Thomas D. Kohler (Reg. No. 32797), Scott D. Stimpson (Reg. No. 33607), Gary S. Williams (Reg. No. 31066), William S. Galliani (Reg. No. 33885), Ann L. Gisolfi (Reg. No. 31956), Todd A. Wagner (Reg. No. 35399), Scott B. Familant (Reg. No. 35514), Kelly D. Talcott (Reg. No. 39582), Francis D. Cerrito (Reg. No. 38100), Anthony M. Insogna (Reg. No. 35203), Brian M. Rothery (Reg. No. 35340), Brian D. Siff (Reg. No. 35679), and Alan Tenenbaum (Reg. No. 34939), all of Pennie & Edmonds LLP, whose addresses are 1155 Avenue of the Americas, New York, New York 10036, 1667 K Street N.W., Washington, DC 20006 and 3300 Hillview Avenue, Palo Alto, CA 94304, and each of them, my attorneys, to prosecute this application, and to transact all business in the Patent and Trademark Office connected therewith.

<b>SEND CORRESPONDENCE TO:</b>		PENNIE & EDMONDS LLP 1667 K Street, N.W. Washington, D.C. 20006		<b>DIRECT TELEPHONE CALLS TO:</b> PENNIE & EDMONDS LLP DOCKETING (202) 496-4721	
201	FULL NAME OF INVENTOR	LAST NAME KAUFFMAN	FIRST NAME STUART	MIDDLE NAME A	
	RESIDENCE & CITIZENSHIP	CITY SANTA FE	STATE OR FOREIGN COUNTRY NEW MEXICO	COUNTRY OF CITIZENSHIP USA	
	POST OFFICE ADDRESS	STREET 1811 S. CAMINO CRUZ BLACA	CITY SANTA FE	STATE OR COUNTRY NEW MEXICO	ZIP CODE 87501
202	FULL NAME OF INVENTOR	LAST NAME TONY	FIRST NAME PLATE	MIDDLE NAME A	
	RESIDENCE & CITIZENSHIP	CITY SANTA FE	STATE OR FOREIGN COUNTRY NEW MEXICO	COUNTRY OF CITIZENSHIP USA	
	POST OFFICE ADDRESS	STREET 812 LOMA BOREAL	CITY SANT FE	STATE OR COUNTRY NEW MEXICO	ZIP CODE 87501
203	FULL NAME OF INVENTOR	LAST NAME GOINS	FIRST NAME TIMOTHY	MIDDLE NAME S.	
	RESIDENCE & CITIZENSHIP	CITY SANTA FE	STATE OR FOREIGN COUNTRY NEW MEXICO	COUNTRY OF CITIZENSHIP USA	
	POST OFFICE ADDRESS	STREET 2543 CAMINO ALFREDO	CITY SANTA FE	STATE OR COUNTRY NEW MEXICO	ZIP CODE 87505
204	FULL NAME OF INVENTOR	LAST NAME COHN	FIRST NAME JUDITH	MIDDLE NAME	
	RESIDENCE & CITIZENSHIP	CITY SANTA FE	STATE OR FOREIGN COUNTRY NEW MEXICO	COUNTRY OF CITIZENSHIP USA	
	POST OFFICE ADDRESS	STREET 325 W. HOUGHTON	CITY SANTA FE	STATE OR COUNTRY NEW MEXICO	ZIP CODE 87501
205	FULL NAME OF INVENTOR	LAST NAME	FIRST NAME	MIDDLE NAME	
	RESIDENCE & CITIZENSHIP	CITY	STATE OR FOREIGN COUNTRY	COUNTRY OF CITIZENSHIP	
	POST OFFICE ADDRESS	STREET	CITY	STATE OR COUNTRY	ZIP CODE
206	FULL NAME OF INVENTOR	LAST NAME	FIRST NAME	MIDDLE NAME	
	RESIDENCE & CITIZENSHIP	CITY	STATE OR FOREIGN COUNTRY	COUNTRY OF CITIZENSHIP	
	POST OFFICE ADDRESS	STREET	CITY	STATE OR COUNTRY	ZIP CODE

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

SIGNATURE OF INVENTOR 201	SIGNATURE OF INVENTOR 202	SIGNATURE OF INVENTOR 203
DATE	DATE	DATE
SIGNATURE OF INVENTOR 204	SIGNATURE OF INVENTOR 205	SIGNATURE OF INVENTOR 206
DATE	DATE	DATE